REMARKS

Claims 1, 3, 5-21, and 23-30 are currently pending. Applicant respectfully requests reconsideration of this application. Claims 1, 8, 10, 18, and 21 have been amended. No claims have been cancelled or added. Therefore, claims 1, 3, 5-21, and 23-30 are now presented for examination.

Claim Rejection under 35 U.S.C. §103

Levine, et al. in view of Mealey, et al.

The Examiner rejected claims 1, 3, 5-21, and 23-30 under 35 U.S.C. 103 (a) as being unpatentable over U.S. Patent 6,134,710 of Levine, et al. (hereinafter referred to as "Levine") in view of U.S. Patent 5,963,737 of Mealey, et al. (hereinafter referred to as "Mealey").

With regard to Levine, the Examiner has indicated that the Examiner does not agree with Applicant's arguments in the prior response. The Examiner states that it was not suggested that the monitor mode control registers of Levine inherently include a handler field to hold a pointer to a handler routine, but rather that Levine teaches a handler routine to process the events. For this reason, the Examiner states that "a pointer to the handler routine is necessary and therefore inherent. Levine is silent as to how or where such a pointer to the handler routine is held." The Examiner states that "Levine does disclose that the monitor mode control registers include fields to control the handler routine (see, for example, bits 5, 16 and 17 in FIG. 3A), suggesting to one of ordinary skill in the art that a similar field could hold the inherent pointer to the handler routine."

It is respectfully submitted that the Examiner has not accurately framed the issue. In response, claim 1 has been clarified to assist in addressing this issue. Claim 1, as amended herein, is as follows:

> 1. An event monitoring component for dynamic optimization comprising:
>
> an event monitor hardware component to selectively capture a plurality of profiles of one or more microarchitecture events occurring in the execution of an application by a microprocessor, the selection of the profiles to be captured and the one or more events to be monitored being based upon configuration information supplied by a software component, the software component including a plurality of handling routines for the processing of captured profiles;
>
> a profile buffer to store the plurality of captured profiles of the one or more microarchitecture events;
>
> an interface through which the software component provides the configuration information to direct the operation of the event monitor; and
>
> one or more monitor control vectors, the monitor control vectors storing the configuration information provided by the software component, wherein each monitor control vector includes a handler field to hold a pointer to a handler routine of the plurality of handling routines, the handler routine being selected by the software component to process the profiles of the microarchitecture event.

Therefore, an event monitoring component includes an event monitor hardware component to selectively capture profiles of one or more microarchitecture events occurring in the execution of an application by a microprocessor, the profiles to be captured and the one or more events to be monitored being based upon configuration

information supplied by a software component. Further, the software component includes a plurality of handling routines. The event monitoring component includes one or more monitor control vectors, with each monitor control vector including a handler field to hold a pointer to a holder routine that is selected by the software component to process the profiles of the microarchitecture event.

Thus, the issue is not simply the existence of a pointer, whether or not the pointer is supported by Levine. The issue is a teaching regarding an event monitoring component that includes the described elements as selected by the software component. Thus, the claim provides a hybrid component that includes the event monitor hardware component, with the events to be monitored, the profiles to be captured, and the handler routine for each monitor control vector being selected as provided in the claim.

Further, it is again submitted that there is no evidence that Levine contemplates a handler field to hold a pointer to handling routine. In this regard, the Examiner indicates that Levine discloses that monitor mode control registers include fields to control a handler routine, specifically citing to bits 5, 16, and 17 of Figure 3A. The Examiner indicates that these suggest to one of skill in the art "that a similar field could hold the inherent pointer to the handler routine". It is submitted this is too speculative to support the existence of any teaching by Levine. It is noted that these are only three bits of a register. As described in Levine, what is illustrated is a performance monitor register, which is capable of being read from and written to under control of a software program. MMCR0 110 controls the operation of PMC0 and PMC1, which are the performance monitoring counters. "Bits 5, 16, and 17 of MMCR0 110 control interrupts generated by PMC0 and PMC1." Bit 5 is an enable bit, but the operation of bits 16 and 17 are unclear.

Thus, it is submitted that Levine does not contain anything that would suggest, explicitly or inherently, the existence of a relevant pointer to a handler routine selected by a software component.

It is submitted that the second cited reference does not contain the elements missing from Levine. Mealey is a patent that concerns an exception handler for a computer system, and specifically regards handling exceptions for performance monitoring. The Examiner has cited such reference for certain aspects with regard to the claims. The Examiner indicates that Mealey expressly discloses a pointer to the address of an interrupt handle, citing to, for example, column 5, line 63 to column 6, line 8. What Mealey indicates is the following:

> In both techniques, the interrupt handlers transfer control to a kernel extension second level interrupt handler which has been previously registered with the kernel. The communication between the kernel extension and the interrupt handlers is via a pointer exported by the kernel. The pointer 32 is a double-word pointer on 64-bit implementations; it is initialized to zero during kernel 16 initialization (e.g., power-up). During kernel extension registration, it is updated with a pointer with the real address 32 of a pinned storage structure 38. The communication structure provides an anchor for ctrace, performance monitor, and for instruction address breakpoint functions and includes a per-processor save area for the TE__FLIH and the IAB__FLIH.

Thus, Mealey is referring to a kernel extension second level interrupt handler that has been previously registered with the kernel. Mealey then goes on to say that a communication between the kernel extension and interrupt handlers is via a pointer exported by the kernel. During kernel extension registration, the pointer is updated with

the real address of a "pinned storage structure 38", the communication structure providing an anchor for ctrace (which, though explained is generally an open-source multithreaded trace/debug library), performance monitor, and instruction address breakpoint functions and includes a save area. While the pointer provides an address to a structure with many tools, what this pointer does NOT point to is a selected handling routine that has been selected by a software component.

The Examiner has further cited to Mealey for the proposition that an arrangement for incorporating a handler field into monitor control registers or vectors such that the handler routine (as of Levine) can be implemented separately for specific applications, citing to, for example, Mealey column 4, lines 15-26. What Mealey describes is a method of handling exceptions that may be tailored to a particular application using a kernel extension. The kernel extension is indicated to be more akin to application software than to the kernel in that it resides in paged memory. In practice, what happens is that when an application signals an exception, the kernel executes code that can lead to an address of a data structure created by the kernel extension that contains the address in the kernel extension to begin executing the exception procedure. (Mealey, col. 3, line 53 to col. 4, line 14) Mealey then indicates that the described exception method has particular utility in performance monitoring tools for microprocessor devices, pointing out that a PowerPC chip has certain performance monitoring hardware built in and that some computer systems employing such a chip use the performance monitoring facilities and some do not. Mealey then indicates that the exceptions needed to make use of the performance monitoring should not be made part of the kernel. (Mealey, col. 4, lines 15-26).

Thus Mealey describes a system that may be used to structure exception systems for different computer environments. Mealey does not address the selection of a handler routine by a software component as provided in claim 1.

Levine and Mealey, alone or in combination, thus do not contain the elements of claim 1, as amended. Independent claims 10 and 21 are also amended, and it is submitted that such claims are allowable for similar reasons. The remaining rejected claims are dependent claims that are allowable as being dependent on the allowable base claims.

Conclusion

Applicant respectfully submits that the rejections have been overcome by the amendment and remark, and that the claims as amended are now in condition for allowance. Accordingly, Applicant respectfully requests the rejections be withdrawn and the claims as amended be allowed.

Invitation for a Telephone Interview

The Examiner is requested to call the undersigned at (503) 439-8778 if there remains any issue with allowance of the case.

Request for an Extension of Time

The Applicant respectfully petitions for a one-month extension of time to respond to the outstanding Office Action pursuant to 37 C.F.R. § 1.136(a). A check is enclosed to cover the necessary fee under 37 C.F.R. § 1.17 for such an extension.

Charge our Deposit Account

Please charge any shortage to our Deposit Account No. 02-2666.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Date: 4/24/06

Mark C. Van Ness
Reg. No. 39,865

12400 Wilshire Boulevard
7th Floor
Los Angeles, California 90025-1026
(303) 740-1980